**Note: Please don't share this document with anyone!**

## Overview

You will build an app with Python and Django. Completing all steps are not necessary but completing each additional step demonstrates how good your problem solving skill is. **Please read the assessment criteria.**

Send us an email if you have a hard time understanding the tasks. We will be putting more details in this doc based on your questions.

Also reply to the mail that you got this link from with the answers. It helps us easily track your progress.

## Assessment Criteria

* The primary assessment criteria is your learning and adapting capability. We are looking for people who can learn a new language, framework etc. in a short period of time and can implement solutions.
* We are also looking for people who can read plain vanilla instructions in english and build systems, as lot of the time work will be remote.
* The more tasks you complete the higher you will rank.
* Clean & Well Documented Code will rank high.
* Early submission will rank high as well.

## Tasks

For each of the steps below, create a git commit with the step name such as “Step 1”, if you have coded up multiple steps at one go, just commit with the latest step that you worked on. Also you can go to the deployment steps if you have completed any of the steps before that; even if you haven't completed all the steps before deployment task, you can deploy whatever you have built. Document in the git readme.md file the packages that you used in each of the steps if you used any, document why you used them.

* Step 1: Build an Auction Site like eBay. In this site anybody can signup using their email address, no complex authentication is needed, when somebody enters their email address at login screen, if the user with the email address already exists it logs the user in and shows a dashboard for the user. If the email address does not exist, a user is created with the email address and the user logs in.
* Step 2: After login, the user will see the auction item gallery, which shows the items everybody else has put up for auction. There should be a create/plus button in the dashboard that allows the user to create an auction item for everybody to see and interact with. If the user hits that button, a form appears, which lets the user input Product Name, Product Description, Product Photo, Minimum Bid Price, and Auction End DateTime. When an auction item is created using this form, it shows up in the auction gallery of everybody else, and shows up in the “My posted items” menu for the posting user.
* Step 3: Users can place bids on items posted by others within the Auction End DateTime. If they click on any item in the auction item gallery, they will be taken to the auction page of that item, where they can see the Product Name, Product Description, Product Photo, Minimum Bid Price, and Auction End DateTime. It will also show a table of bids placed by other users for that product. An user can input their bid in the auction item page for that product. After inputting a bid, it shows up in the bid list. The user can edit their bid before the auction ends.
* Step 4: The auction will end at Auction End DateTime, If you enter the auction item page for any item, it will show the bid winner for that item.
* Step 5: Make an admin dashboard where the admin can see the auction gallery and auction item pages. Also the admin can see statistics on how many total auctions are running now, what is their total value. Include a chart where it shows a time series of auction added count, and auction completed count in two variables,add another chart that shows the total auction value by time, based on the latest bid placed. You should be able to change the frequency of these charts from a dropdown, frequencies are minute, hour and day. The admin can login at /admin endpoint, with the default admin/admin user pass.
* Step 6: Make everything look pretty based on your judgment.
* Step 7: Whatever you have built, deploy it in heroku and send us the link at hr@taka.school. Also put this on github and send us a repo link as well. Reply to the mail that you were sent the task in .
* Step 8: Put in the readme.md file all the challenges that you have faced in each of the steps, and how you solved them.